Stiffness in Stochastic Chemically Reacting Systems: The Implicit Tau-Leaping Method

Muruhan Rathinam* Linda R. Petzold† Yang Cao‡ Daniel T. Gillespie§

August 19, 2003

Abstract

We show how stiffness manifests itself in the simulation of chemical reactions at both the continuous-deterministic level and the discrete-stochastic level. Existing discrete stochastic simulation methods, such as the Stochastic Simulation Algorithm and the (explicit) tau-leaping method, are both exceedingly slow for such systems. We propose an implicit tau-leaping method that can take much larger time steps for many of these problems.

1 Introduction

In microscopic systems formed by living cells, the small numbers of reactant molecules can result in dynamical behavior that is discrete and stochastic rather than continuous and deterministic.[1–4] An analysis tool that respects these dynamical characteristics is the stochastic

*Department of Mathematics and Statistics, University of Maryland Baltimore County, muruhan@math.umbc.edu

†Computational Science and Engineering, University of California Santa Barbara petzold@engineering.ucsb.edu

‡Computational Science and Engineering, University of California Santa Barbara ycao@engineering.ucsb.edu

§Dan T. Gillespie Consulting, GillespieDT@mailaps.org

simulation algorithm (SSA), a numerical simulation procedure that is essentially exact for chemical systems that are spatially homogeneous or well stirred. Despite recent improvements,[5] as a procedure that simulates *every* reaction event, the SSA is necessarily inefficient for most realistic problems. There are two main reasons for this, both arising from the multi-scale nature of the underlying problem: (1) *stiffness*, i.e. the presence of multiple timescales, the fastest of which are stable; and (2) the need to include in the simulation both species that are present in relatively small quantities and should be modeled by a discrete stochastic process, and species that are present in larger quantities and are more efficiently modeled by a deterministic differential equation (or at some scale in between). We emphasize that most chemical systems, whether considered at a scale appropriate to stochastic or to deterministic simulation, involve several widely varying time scales, so such systems are *nearly always stiff*.

In this paper we will address the problem of stiffness for discrete stochastic systems. We will demonstrate how stiffness is manifested in stochastic chemical kinetics, and show how to modify the recently-proposed tau-leaping method[6] so that much longer timesteps can be taken for stiff systems.

The SSA, the tau-leaping method,[6] the modified tau-leaping method that will be introduced here, and deterministic ordinary differential equation (ODE) simulation are each most effective in certain situations. When the populations of all reactant species are small, the SSA will be as fast and efficient as one could reasonably wish. The goal of tau-leaping was to speed up the SSA when either all reactant species are present in moderately large numbers, or (much more commonly) when some reactant species are present in small or moderate numbers while others are present in very large numbers. In such situations, significant stochastic effects can still arise, but tracking them with the SSA will be very time consuming. These situations can be expected to arise in many cellular systems of interest to biochemists. For those systems, the exact SSA is usually much too slow, while the deterministic reaction rate equation (RRE),

though fast, fails to capture the stochastic effects. It was shown in[6] that tau-leaping morphs into the SSA when all the molecular populations are very small, and morphs into the explicit Euler method for the RRE when all the molecular populations are very large. Our present work is aimed at formulating tau-leaping strategies that accurately and efficiently handle systems "in between" those two extremes.

The outline of this paper is as follows. In Section 2 we review simulation algorithms for chemical kinetics for a wide range of scales. In Section 3 we outline the problem of stiffness for the simulation of chemical kinetics systems at both the continuous and stochastic scales. In Section 4 we propose an *implicit tau-leaping method* that overcomes the stepsize limitations due to stiffness of the explicit tau-leaping method, and we outline its implementation. Finally, in Section 5 we present some numerical experiments demonstrating both the effectiveness and some limitations of the new implicit tau-leaping method.

2 Simulation Algorithms for Chemical Kinetics

In a chemically reacting system involving N molecular species $\{S_1, \ldots, S_N\}$, the state vector $\mathbf{X}(t) \equiv (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the number of molecules of species S_i in the system at time t, will evolve stochastically because of the inherent randomness of thermal molecular motion. Stochastic molecular collisions give rise to stochastic chemical transmutations in accordance with a given set of M reaction channels $\{R_1, \ldots, R_M\}$. If the system is well-stirred and in thermal equilibrium, the dynamics of reaction channel R_j will be completely characterized by a *propensity function* a_j and a *state-change vector* $\boldsymbol{\nu}_j = (\nu_{1j}, \ldots, \nu_{Nj})$: $a_j(\mathbf{x})dt$ gives the probability that one R_j reaction will occur in state $\mathbf{x}$ during the next infinitesimal time interval dt, and ν_{ij} gives the change in the S_i molecular population induced by one R_j reaction.[7]

By appealing to the laws of probability theory, one can derive a chemical master equation

(CME) that governs the time evolution of the probability density function of $\mathbf{X}(t)$, as well as a stochastic simulation algorithm (SSA) that can generate numerical realizations of $\mathbf{X}(t)$. Both the CME and the SSA are exact consequences of the foregoing dynamical assumptions, so in spite of the difference in their descriptive thrusts, they are logically equivalent to each other.

The SSA simulates each successive reaction event that occurs in the system. It is a Monte Carlo method which proceeds from the fact that, if $\mathbf{X}(t) = \mathbf{x}$, then with

$$a_0(\mathbf{x}) \triangleq \sum_{j=1}^{M} a_j(\mathbf{x}),$$

the time τ to the next reaction event is an exponentially distributed random variable with mean $1/a_0(\mathbf{x})$, and the index j of that next reaction is an integer random variable with probability $a_j(\mathbf{x})/a_0(\mathbf{x})$. Because the SSA simulates one reaction at a time, it will be very slow in the commonly occurring case that some reactions take place on a very fast timescale. Although exact methods have been proposed[5] that speed up the SSA, by itself it remains much too slow for practical simulation of realistic biological systems.

An *approximate* scheme called *tau-leaping* has recently been proposed[6] to accelerate the SSA. The basic idea of tau-leaping is as follows. Given a *pre-selected* time step τ that encompasses more than one reaction event, if we could determine how many times each reaction channel fired during that time step, we might be able to forego knowing the precise instants at which those firings took place. In such a circumstance, we could leap along the system's history axis from one τ-subinterval to the next, instead of stepping along from one reaction event to the next. It has been shown[6] that this can be done *approximately* if τ is taken small enough that the propensity functions remain nearly constant during the time step. The tau-leaping simulation method is an attempt to speed up the SSA by sacrificing some exactness. But the approximate method must be used with circumspection, since while we are glad to leap over "unimportant" reaction events, we must take care not to leap over "important" ones.

To render these ideas more precisely, in tau-leaping attention is focused on the set of M

random variables

$$K_j(\tau;\mathbf{x},t) \equiv \text{the number of times reaction channel } R_j \text{ fires in } [t,t+\tau), \text{ given that } \mathbf{X}(t)=\mathbf{x}, \quad (j=1,\ldots,M). \tag{1}$$

It follows from the above definitions that if the system is in state $\mathbf{x}$ and reaction R_1 fires k_1 times, and reaction R_2 fires k_2 times, etc., then the system will change to state $\mathbf{x}+\sum_{j=1}^{M} k_j\boldsymbol{\nu}_j$. Therefore, the random variables $K_j(\tau;\mathbf{x},t)$ defined in (1) completely determine the evolution of the system as follows: If $\mathbf{X}(t)=\mathbf{x}$, then for any $\tau>0$,

$$\mathbf{X}(t+\tau)=\mathbf{x}+\sum_{j=1}^{M} K_j(\tau;\mathbf{x},t)\boldsymbol{\nu}_j. \tag{2}$$

The simple (explicit) tau-leaping method makes the approximation

$$K_j(\tau;\mathbf{x},t) \approx \mathcal{P}_j(a_j(\mathbf{x}),\tau)$$

where the $\mathcal{P}_j$ are statistically independent Poisson random variables.[8] Thus by (2), the explicit tau-leaping algorithm takes the following form: If $\mathbf{X}(t)=\mathbf{x}$, then for any $\tau>0$,

$$\mathbf{X}(t+\tau) \approx \mathbf{x}+\sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}_j(a_j(\mathbf{x}),\tau). \tag{3}$$

It has been shown[6] that this tau-leaping method limits to the SSA-method as the time step τ becomes smaller than the mean time to the next reaction. In a forthcoming paper, we will present an analysis of this tau-leaping method, which in particular shows that the method is first order accurate in τ.

At the next coarser scale, suppose conditions are such that, starting in state $\mathbf{x}$ at time t, we can leap over an interval τ that spans a *very large* number of firings of *every* reaction channel, yet all those firings induce only miniscule changes in the values of all the propensity functions. Then, since the Poisson random variable $\mathcal{P}(a,t)$ will, when $at >> 1$, be well approximated by the *normal* random variable $\mathcal{N}(at,at)$,[9] the number of firings of channel R_j in $[t,t+\tau)$ can be

approximated by

$$\begin{aligned} K_j(\tau;\mathbf{x},t) &\approx \mathcal{P}_j(a_j(\mathbf{x}),\tau) \\ &\approx \mathcal{N}_j(a_j(\mathbf{x})\tau, a_j(\mathbf{x})\tau) \\ K_j(\tau;\mathbf{x},t) &\approx a_j(\mathbf{x})\tau + (a_j(\mathbf{x})\tau)^{1/2}\mathcal{N}_j(0,1). \end{aligned}$$

Substituting this into Equation (2) yields the *Langevin method*: If $\mathbf{X}(t) = \mathbf{x}$, then for any $\tau > 0$,

$$\mathbf{X}(t+\tau) \approx \mathbf{x} + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{x}) + \tau^{1/2} \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j^{1/2}(\mathbf{x}) \mathcal{N}_j(0,1), \tag{4}$$

where the $\mathcal{N}_j(0,1)$ are statistically independent normal random variables with means 0 and variances 1. Equation (4) is in fact the well known first order explicit method for simulating the *continuous* Markov process defined by the *chemical Langevin equation.*[10,11] A great deal of work in the past decade has gone into developing theory and numerical methods for equations of this type, which are known in the mathematical literature as stochastic differential equations (SDEs). Well-developed theory exists for determining the order of convergence of this and higher order methods for SDEs.[12] Some recent work has addressed automatic stepsize selection.[13]

Finally, in the limit of *infinitely* large molecular populations of all of the reactant species, or more specifically in the *thermodynamic limit*, each term in the second summation on the right hand side of (4) usually becomes vanishingly small compared to the correspondingly indexed term in the first summation.[14] Therefore, in that limit Equation (4) usually reduces to, again with $\mathbf{X}(t) = \mathbf{x}$,

$$\mathbf{X}(t+\tau) \approx \mathbf{x} + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{x}). \tag{5}$$

This will be recognized as the *explicit Euler method* for the numerical solution of the deterministic ODE system given by the *reaction rate equations* (which are more commonly scaled by the system volume).

3 Stiffness

In *deterministic* systems of ODEs, stiffness generally manifests when there are well separated "fast" and "slow" time scales present, and the "fast modes" are stable. Because of the fast stable modes, all initial conditions result in trajectories which, after a short and rapid transient, lead to the "stable manifold" where the "slow modes" determine the dynamics and the fast modes have decayed.

In general, a given trajectory of such a system will exhibit rapid change for a short duration (corresponding to the fast time scales) called the "transient", and then evolve slowly (corresponding to the slow time scales). During the initial transient the problem is said to be nonstiff, whereas while the solution is evolving slowly it is said to be stiff. One would expect that a reasonable numerical scheme should be able to take larger time steps once the trajectory has come sufficiently close to the slow manifold without compromising the accuracy of the computed trajectory. That this is not always the case is well known to numerical analysts, and in general explicit methods are only able to perform well if they continue to take time steps that are of the order of the fastest time scale. This happens because explicit methods advance the solution from one time to the next by approximating the slope of the solution curve at or near the beginning of the time interval. Since any numerical method makes errors on every time step, the numerical solution is never exactly on the stable manifold. Instead, it will be on some trajectory that approaches the stable manifold very rapidly. Thus the approximation to the slope employed by explicit methods will always be on the order of the fastest time scale of the system. The very large slope decreases to almost zero in a time interval of the order of the fastest time scale. If the explicit numerical scheme continues to take small time steps of the order of these fast trajectories, then there is no problem. However, if the explicit method takes a larger time step, which would seemingly be appropriate for following the trajectories on the slow manifold, then the large estimated slope and the large time step lead to a point

on the other side of the the slow manifold, which is likely to be further away from it than was the previous point. This point is likely to have an even larger slope, leading to highly unstable oscillations.

An implicit method on the other hand does not approximate the slope of the trajectory near the beginning of the interval of a time step. Instead, it gives more weight to the slope at the unknown point at the end of the time step. This tends to avoid the above described instability, but at the expense of having to solve a nonlinear system of equations for the unknown point at each time step. In fact, implicit methods often damp the perturbations off the slow manifold. Once the solution has reached the stable manifold, this damping keeps the solution on the manifold, and is desirable. Further details on stiffness in deterministic ODE systems can be found in[15] and the references therein.

The aim of this paper is to explore the nature of stiffness in discrete stochastic systems, to propose an implicit version of the (explicit) tau-leaping method discussed in Section 2, and to demonstrate the extent to which the implicit method is effective for stiff discrete stochastic systems.

When stochasticity is introduced into a system with fast and slow time scales, with fast modes being stable as before, one may still expect a slow manifold corresponding to the equilibrium of the fast scales. But the picture changes in a fundamental way. After an initial rapid transient, while the mean trajectory is almost on the slow manifold, *any sample trajectory will still be oscillating at the fast time scale in a direction transverse to the slow manifold.* In some cases the size of the fluctuations off the slow manifold will be practically negligible. In those circumstances, an implicit scheme may take large steps, corresponding to the time scale of the slow mode. But in other cases, the fluctuations off the slow manifold will *not* be negligible in size. In those instances, an implicit scheme that takes time steps much larger than the time scale of the fast dynamics will dampen these fluctuations, and will consequently

fail to capture the variance correctly.

We will demonstrate that the implicit tau-leaping method can take large time steps for stiff discrete stochastic systems, producing a solution which is accurate for the slow variables of the system, and for which the mean of the fast variables on the slow manifold is accurate. We will also show how the distribution of the fast variables on the slow manifold can be recovered at relatively low cost.

4 The Implicit Tau-Leaping Method

The tau-leaping method described by (3) is an explicit method because the propensity functions a_j are evaluated at the current known state, so the future unknown random state $\mathbf{X}(t+\tau)$ is given as an explicit function of $\mathbf{X}(t)$. Throughout the rest of the paper we shall refer to (3) as the *explicit-tau* method, and write it

$$\mathbf{X}^{(et)}(t+\tau) = \mathbf{X}^{(et)}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}_j(a_j(\mathbf{X}^{(et)}(t)), \tau), \tag{6}$$

where the superfix "et" stands for explicit-tau. We mentioned in Section 3 that the explicit Euler method exhibits instability for stiff systems with large stepsizes. The explicit-tau method is essentially an extension of the explicit Euler method to discrete stochastic systems, and as such it too has poor stability. In this section we motivate and derive an implicit tau-leaping method. In Section 5 we will present numerical experiments that demonstrate the accuracy and efficiency of this method, as compared to the explicit-tau method and the SSA.

To motivate our formulation of the implicit-tau method, we look again at the explicit-tau method (6). Here the increment in the state $\mathbf{X}^{(et)}(t+\tau) - \mathbf{X}^{(et)}(t)$ is given by a linear combination of statistically independent Poisson random variables $\mathcal{P}_j(a_j, \tau)$ whose parameters a_j are evaluated at $\mathbf{X}^{(et)}(t)$. An attempt to completely implicitize the method would require generating Poisson random variables $\mathcal{P}_j(a_j, \tau)$ with the a_j evaluated at the unknown random

state $\mathbf{X}(t+\tau)$ that we are trying to find. Since it is not entirely clear how to interpret and solve such an equation, we will attempt a partial implicitization. To this end, let us regard each of the random variables $\mathcal{P}_j$ as the sum of two parts, one being the mean value $a_j\tau$ of $\mathcal{P}_j$, and the other being the zero-mean random variable $\mathcal{P}_j - a_j\tau$. We then evaluate the mean value part $a_j\tau$ at the unknown state $\mathbf{X}(t+\tau)$, and the zero-mean random part $\mathcal{P}_j - a_j\tau$ at the known state $\mathbf{X}(t)$. Thus we arrive at an implicit method described by

$$\begin{aligned} \mathbf{X}^{(it)}(t+\tau) &= \mathbf{X}^{(it)}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}^{(it)}(t+\tau))\,\tau \\ &+ \sum_{j=1}^{M} \boldsymbol{\nu}_j \left(\mathcal{P}_j(a_j(\mathbf{X}^{(it)}(t)),\tau) - a_j(\mathbf{X}^{(it)}(t))\,\tau \right). \end{aligned} \tag{7}$$

Here the random variables $\mathcal{P}_j$ are, as before, statistically independent Poisson random variables. In a forthcoming paper, we will present an analysis of the accuracy and stability properties of this method, which in particular shows that the method is accurate to first order in τ.

We note that in the implementation of the method (7), the random variables $\mathcal{P}_j(a_j(\mathbf{X}^{(it)}(t)),\tau)$ can be generated without knowing $\mathbf{X}^{(it)}(t+\tau)$. Also, once the $\mathcal{P}_j(a_j(\mathbf{X}^{(it)}(t)),\tau)$ have been generated, the unknown state $\mathbf{X}^{(it)}(t+\tau)$ depends on $\mathcal{P}_j(a_j(\mathbf{X}^{(it)}(t)),\tau)$ in a deterministic way, even though this dependence is given by an implicit equation. As is done in the case of deterministic ODE solution by implicit methods, $\mathbf{X}^{(it)}(t+\tau)$ can be computed by applying Newton's method for the solution of nonlinear systems of equations to (7) where the $\mathcal{P}_j(a_j(\mathbf{X}^{(it)}(t)),\tau)$ are all known values.

Just as the explicit-tau method segues to the explicit Euler methods for SDEs and ODEs, the implicit-tau method segues to the implicit Euler methods for SDEs and ODEs. In the SDE regime we get, approximating Poissons by normals

$$\mathbf{X}^{(it)}(t+\tau) \approx \mathbf{X}^{(it)}(t) + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}^{(it)}(t+\tau)) + \tau^{1/2} \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}^{(it)}(t))^{1/2} \mathcal{N}_j(0,1), \tag{8}$$

where $\mathcal{N}_j(0,1)$ are independent normal random variables with mean zero and variance 1. This

is precisely the implicit Euler version of (4).[12]

In the thermodynamic limit[14] where random terms in the above SDE system may be ignored, the implicit-tau method becomes the implicit Euler method

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \tau \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}(t+\tau)), \tag{9}$$

for the corresponding deterministic reaction rate equations.

It is well known that, for stiff ODE systems, the implicit Euler method has a strong damping property. Indeed, it is this property that makes the implicit Euler method so desirable for such systems: once the solution is close enough to the slow manifold that the stepsize can be increased, the method damps out any errors and keeps the solution close to the slow manifold. The implicit tau-leaping method inherits this damping property, which is still advantageous for taking large time steps and staying close to the slow manifold. However, as a consequence of this property, the method will also damp out the natural fluctuations of the fast variables. So while the implicit tau-leaping method computes the *slow* variables with their correct distributions, it computes the *fast* variables with the correct means but with distributions about those means that are too narrow.

We have developed a time-stepping strategy that is intended to *restore* the overly-damped fluctuations in the fast variables. The idea is to interlace the implicit tau-leaps, each of which is on the order of the time scale of the slow variables and hence "large", with a sequence of many much smaller time steps, each of which is on the order of the time scale of the fast variables. The smaller time steps are to be taken over a duration that is comparable to the "relaxation/decorrelation" time of the fast variables. These small time steps may be executed using either the explicit-tau or the implicit-tau. This sequence of small steps is intended to "regenerate" the correct statistical distributions of the fast variables, which have been made too narrow by the preceding large implicit tau-leaps. The fact that the underlying kinetics is Markovian or "past-forgetting" is important in being able to apply such a procedure. The

optimal interlacing strategy and the choice of explicit versus implicit-tau for the small time steps is the subject of further research.

In the next section, our first example will illustrate the damping of the fluctuations in the fast variables caused by successive large implicit-tau leaps, and then the successful regeneration of those fluctuations through a sequence of 10 successive small implicit-tau leaps, all with a very substantial net gain in computational efficiency.

Finally, we note that the implicit-tau method (7) has the property that the state change $\mathbf{X}^{(\mathrm{it})}(t+\tau) - \mathbf{X}^{(it)}(t)$ is generally not an integer vector. It is possible to avoid non-integer state changes by modifying the implicit-tau method. It might be tempting to do this by simply rounding every component of $\mathbf{X}^{(\mathrm{it})}(t+\tau)$ to the nearest integer. But it is better to ensure that the state change be *stoichiometrically realizable*; i.e., not only should the state change be an integer vector, but it should also be a sum of the form $k_1\boldsymbol{\nu}_1 + \cdots + k_M\boldsymbol{\nu}_M$, where $k_1, \ldots, k_M$ are nonnegative integers. This way, the state change can be interpreted as the result of reaction channel R_j firing k_j times for $j = 1, \ldots, M$. This yields the following implicit method: First, compute $\mathbf{X}' = \mathbf{X}(\mathbf{t} + \tau)$ according to (7), i.e. by using Newton's method to solve the implicit equation

$$\mathbf{X}' = \mathbf{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X}')\,\tau + \sum_{j=1}^{M} \boldsymbol{\nu}_j \left(\mathcal{P}_j(a_j(\mathbf{x}), \tau) - a_j(\mathbf{x})\,\tau\right), \tag{10}$$

where $\mathbf{x}$ is the system's state at time t. Then approximate the number of firings $K_j(\tau; \mathbf{x}, t)$ of the reaction channel R_j in the time interval $[t, t+\tau]$ by the *integer-valued* random variable $\hat{K}_j(\tau; \mathbf{x}, t)$ defined by

$$\hat{K}_j(\tau; \mathbf{x}, t) = \left[a_j(\mathbf{X}')\,\tau + \mathcal{P}_j(a_j(\mathbf{x}), \tau) - a_j(\mathbf{x})\,\tau\right]. \tag{11}$$

Here the $\mathcal{P}_j(a_j(\mathbf{x}), \tau)$ for $j = 1, \ldots, M$ are the *same numbers* used in equation (10), and $[z]$ denotes the nearest nonnegative integer corresponding to a real number z. Finally, invoking

(2), estimate the state at time $t+\tau$ as

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j \hat{K}_j(\tau; \mathbf{x}, t). \tag{12}$$

Although this modification might be prudent for some systems, for the simple systems we have studied thus far, the original version of the implicit tau method has performed as well as this "rounded" version. Therefore, in the remainder of this paper we will focus on the original "unrounded" implicit-tau method.

5 Numerical Experiments

Example 1 This problem, the decaying-dimerizing reaction set studied in,[6] consists of three species S_1, S_2 and S_3 and four reaction channels:

$$\begin{aligned} S_1 &\xrightarrow{c_1} 0 \\ S_1 + S_1 &\xrightarrow{c_2} S_2 \\ S_2 &\xrightarrow{c_3} S_1 + S_1 \\ S_2 &\xrightarrow{c_4} S_3. \end{aligned} \tag{13}$$

We chose values for the parameters

$$c_1 = 1,\ c_2 = 10,\ c_3 = 1000,\ c_4 = 0.1$$

which will render the problem stiff. The initial conditions $x_1(0) = 400$, $x_2(0) = 798$ and $x_3(0) = 0$ were chosen to lie on the approximate slow manifold given by the equation

$$x_2 = \frac{5}{1000} x_1(x_1 - 1).$$

This avoids the inconvenience for the constant-stepsize algorithms under study of having to take small steps during the initial transient, and large steps on the slow manifold. The propensity functions are given by

$$a_1 = x_1,\ a_2 = 5x_1(x_1 - 1),\ a_3 = 1000x_2,\ a_4 = 0.1x_2,$$

and the problem was solved on the time interval $[0, 0.2]$.

Figure 1 depicts sample trajectories as simulated by the exact SSA. Figure 2 shows the same sample trajectory of x_3, on a more revealing scale for that variable. We note that while x_1 and x_2 vary rapidly, x_3 varies slowly. All three variables x_1, x_2 and x_3 exhibit random behavior, x_3 being the most random. Figures 3, 4 and 5 show the histograms for the final state values, comparing SSA with explicit tau-leaping. Each histogram was obtained by simulating an ensemble of $10,000$ trajectories. The explicit tau-leaping was performed with a constant stepsize of 2×10^{-5}.

It is evident from Figures 3, 4 and 5 that the explicit-tau method captures the statistics of the final states very well with only $10,000$ time steps over the interval, whereas the SSA required on average $310,000$ time steps. In terms of computation time, $10,000$ simulations using SSA took $5,697$ cpu seconds, while $10,000$ simulations using explicit-tau with a constant stepsize of 2×10^{-5} took 731 cpu seconds. These computations were performed on a 1.4Ghz Pentium IV Linux workstation.

Figure 6 shows that the explicit-tau method becomes unstable at stepsizes roughly equal to or larger than 2.2×10^{-4}. This is the stability limit that would be predicted by a linearized stability analysis of the forward Euler method applied to the corresponding deterministic ODE model. In a forthcoming paper, we will address stability criteria and analysis for discrete stochastic systems.

To verify that the implicit-tau method can take much larger time steps while maintaining accuracy, we simulated an ensemble of $10,000$ trajectories using explicit-tau with constant stepsize 10^{-4} and implicit-tau with constant step size 0.01. The stepsize 10^{-4} for explicit-tau was chosen to be as large as possible without compromising accuracy (it is near the stability limit of 2.2×10^{-4}). Figures 7, 8 and 9 compare the final state relative bin frequencies computed by all three methods. Tables 1 and 2 show the sample means and standard deviations for the

final states estimated by the three methods.

	SSA	Explicit-Tau	Implicit-Tau	Interlaced Implicit-Tau
Sample mean : $x_1(0.2)$	387.3	386.2	387.6	387.2
Sample mean : $x_2(0.2)$	749.5	750.1	749.5	749.4
Sample mean : $x_3(0.2)$	15.45	15.48	15.42	15.59

Table 1: Sample means (for a sample size of $n = 10,000$) for the final states in Example 1 as computed by SSA, explicit-tau, the original implicit-tau, and the interlaced implicit-tau methods, with stepsizes as described in the text.

	SSA	Explicit-Tau	Implicit-Tau	Interlaced Implicit-Tau
Sample standard deviation : $x_1(0.2)$	18.42	24.76	3.07	17.74
Sample standard deviation : $x_2(0.2)$	10.49	13.45	5.34	10.24
Sample standard deviation : $x_3(0.2)$	3.91	3.88	3.89	3.91

Table 2: Sample standard deviations (for a sample size of $n = 10,000$) for the final states in Example 1 as computed by SSA, explicit-tau, the original implicit-tau, and the interlaced implicit-tau methods, with stepsizes as described in the text.

It is clear from the histograms that if the goal is to capture the slow state x_3 including its randomness (which is significant), then implicit-tau is far superior to explicit-tau, because it achieves comparable accuracy with a factor of 100 fewer steps. Although the computational effort per step is greater for implicit-tau than explicit-tau, this is far out-weighed by implicit-tau's ability to take much larger steps. The computation time for $10,000$ implicit-tau simulations was 15 cpu-seconds, as compared to 731 cpu-seconds for explicit-tau and $5,697$ cpu-seconds for SSA.

On the other hand, if one is interested in capturing the fluctuations of the fast variables x_1 and x_2, then implicit-tau with these large time steps will not be adequate. In order to capture

the distributions of x_1 and x_2, we used the technique described in Section 4 of interlacing small time steps with large time steps. In this example, we took the first 19 steps with implicit-tau using stepsize 0.01 as before. Then implicit-tau was used to take one step of size 0.0098. For the remaining time of 0.002 we took 10 steps of size 2×10^{-5} using implicit-tau. The first 20 steps are the large steps which capture the *mean* values of all the state variables, and the noise in the slow variable, accurately. The last 10 small steps recover the distribution information of the fast variables x_1 and x_2. The time period to recover the distribution information is roughly the "relaxation time" of the fast variables. The final state histogram for this simulation is compared with that of the SSA in Figures 10, 11 and 12, and shows good agreement with SSA for all three variables x_1, x_2 and x_3. The computation time for $10,000$ simulations using this interlaced method was 23 cpu-seconds (versus 731 cpu-seconds for the explicit-tau simulations in Figures 3, 4 and 5). In this example, we were able to capture the distributions of all state variables at the final time with only one recovery period. In general, it may be necessary to do the recovery steps more often. An optimal strategy for the interlacing procedure is a topic for further research.

Example 2 In some stiff systems, the fast variables exhibit near deterministic behavior while the slow variables still exhibit randomness. In such cases, implicit-tau will clearly be the method of choice, as can be demonstrated using the simple reaction set

$$\begin{aligned} R_1: &\quad S_1 \xrightarrow{c_1} S_3, \\ R_2: &\quad S_3 \xrightarrow{c_2} S_1, \\ R_3: &\quad S_1 + S_2 \xrightarrow{c_3} S_1 + S_4. \end{aligned}$$

Since the total number of S_1 and S_3 molecules is constant (say x_T), and if we don't care about the byproduct S_4, we can model this system with two variables $x = (x_1, x_2)$ (i.e. numbers

of S_1 and S_2 molecules respectively) and three reactions. The propensity functions are

$$a_1(x) = c_1 x_1$$

$$a_2(x) = c_2(x_T - x_1)$$

$$a_3(x) = c_3 x_1 x_2.$$

The stoichiometric vectors are $\nu_1 = (-1, 0)^T$, $\nu_2 = (1, 0)^T$, and $\nu_3 = (0, -1)^T$. We chose $c_1 = c_2 = 10^5$, $c_3 = 0.0005$, and $x_T = 20,000$, with initial condition $x(0) = (10000, 100)$.

Since $c_1 = c_2$, then $x_1 = x_T/2$ will be an equilibrium value for x_1. The dynamics of x_1 is independent of x_2, but the dynamics of x_2 depends on x_1. Also note that the reactions R_1 and R_2 are much faster than R_3.

Since the dynamics of x_1 alone is the same as in the simple reversible isomerization problem,[11] the exact asymptotic mean and variance can be computed analytically (see the Appendix). The equilibrium value $x_1 = 10,000$ is the asymptotic mean, and the asymptotic standard deviation of x_1 is given by $\sqrt{\frac{c_1 x_\infty}{c_1 + c_2}} = \sqrt{10000/2} = 70.7$ (here x_∞ is the asymptotic mean value, which in our case is $x_T/2$). This is less than 1% of the equilibrium value. Thus we may regard the noise in x_1 as negligible. But as we shall see, the noise in x_2 cannot be regarded as negligible.

We simulated an ensemble of $10,000$ trajectories using all three methods: SSA, explicit-tau with constant step size 5×10^{-6} (half the size of the maximum value to maintain stability), and implicit-tau with constant step size 0.005, all to estimate the final state at time $T = 0.01$.

Figures 13 and 14 compare the final state histograms computed by explicit-tau and implicit-tau with those computed by SSA. The full behavior of the noisy variable x_2 is adequately reproduced by both tau-leaping methods, while the inaccuracies of both methods in the estimations of the fluctuations in x_1 are inconsequential because of their smallness. The implicit-tau method is superior to the explicit-tau method, since the former takes 2 steps for each trajectory while the latter takes $2,000$. The explicit-tau method in turn is superior to the SSA,

which takes on average 2×10^7 time steps for each trajectory.

Note that in this reaction, one can make the relative size of the equilibrium noise in x_1 arbitrarily small by scaling up c_1 and c_2 and $x_1(0) = x_T/2$ by the same factor. This leaves the stiffness ratio $\frac{2(c_1+c_2)}{c_3 x_T}$ unchanged but makes the noise in x_1 as small as we want compared to its equilibrium value. For instance if we choose $c_1 = c_2 = 10^6$ and $x(0) = (10^6, 100)$ then the equilibrium noise of x_1 will have a standard deviation of $\sqrt{50000} \approx 224$, which is 0.02% of the equilibrium value 10^6. Thus the noise will be less than what we obtained with our choice for c_1, c_2 and x_T. We did not choose the values $c_1 = c_2 = 10^6$ and $x_T = 200,000$ because the SSA simulation takes an extremely long time to run and we wanted an example where we could make a quantitative comparison with SSA.

6 Conclusions

We have shown how stiffness manifests itself in the simulation of chemical reactions at both the continuous, deterministic level and the discrete, stochastic level. While the explicit-tau method is an important first step in the efficient simulation of stochastic chemical systems, it must use a very small stepsize when applied to stiff systems.

We have proposed an implicit version of the tau-leaping method. We have demonstrated through numerical simulations that the implicit tau method achieves the same level of accuracy as the explicit-tau method when the latter is stable, and that the new method overcomes the instability problem of explicit-tau for larger stepsizes. For large stepsizes, we have seen that the implicit tau method resolves well the slow stochastic components, and it captures the mean of the fast components. We have introduced a method for recovering the distributions of the fast stochastic components based on a time stepping scheme that interlaces several small time steps with several large time steps.

Acknowledgements

We would like to thank John Doyle for bringing us together and for making us aware of the need for multiscale methods in the simulation of biochemical networks. We would also like to thank Andrew Hall for assisting us with the numerical experiments. This work was supported by the Air Force Office of Scientific Research and the California Institute of Technology under DARPA Award No. F30602-01-2-0558.

References

[1] H.H. McAdams and A. Arkin. It's a noisy business! *Trends in Genetics*, 15(2):65–69, Feb 1999.

[2] H.H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, 94:814–819, 1997.

[3] A. Arkin, J. Ross, and H.H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *e. coli* cells. *Genetics*, 149:1633–1648, Aug 1998.

[4] Nina Fedoroff and Walter Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 16 Aug 2002.

[5] M. A. Gibson and J. Bruck. Exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.*, 105:1876–1889, 2000.

[6] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115(4):1716–1733, 2001.

[7] As here defined, ν_{ij} is the ν_{ji} of Refs.[6,10,11] The present indexing corresponds to the commonly accepted definition of the *stoichiometric matrix*.

[8] The Poisson random variable $\mathcal{P}(a,\tau)$ is the integer-valued random variable defined by

$$\text{Prob}\{\mathcal{P}(a,\tau)=n\} = \frac{e^{-a\tau}(a\tau)^n}{n!} \quad (n=0,1,\ldots).$$

Both the mean and the variance of $\mathcal{P}(a,\tau)$ are equal to $a\tau$. $\mathcal{P}(a,\tau)$ can be interpreted physically as the number of events that will occur in any finite time τ, given that the probability of an event occurring in any future *infinitesimal* time dt is $a\,dt$.

[9] We denote by $\mathcal{N}(m,\sigma^2)$ the normal (or Gaussian) random variable with mean m and variance σ^2. This random variable has the useful property that $\mathcal{N}(m,\sigma^2) = m + \sigma\mathcal{N}(0,1)$.

[10] D. T. Gillespie. The chemical Langevin equation. *J. Chem. Phys.*, 113:297–306, 2000.

[11] D. T. Gillespie. The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction. *J. Phys. Chem. A*, 106:5063–5071, 2002.

[12] P. F. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer Verlag, second edition, 1995.

[13] P. M. Burrage and K. Burrage. A variable stepsize implementation for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(3):848–864, 2002.

[14] In the *thermodynamic limit*, the species populations x_i and the system volume Ω diverge *together*, and *proportionately*. It turns out that, in this limit, all propensity functions $a_j(\mathbf{x})$ diverge *linearly* with the system size; because, the propensity function for an m-th order reaction will contain m factors x_i along with a factor $\Omega^{-(m-1)}$. As a consequence, while the terms under the first summation sign in Eq. 4 are roughly proportional to the system size, the terms under the second summation sign are roughly proportional to the *square root* of the system size. So in the thermodynamic limit, the latter terms typically become negligibly small compared to the former terms. Of course, real systems, no matter how large, are necessarily *finite*, and in situations where the terms in the first summation in Eq. 4 add up to practically zero (for instance at equilibrium), the fluctuating second sum can become important .

[15] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM, 1998.

[16] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists.* Academic Press, 1992.

[17] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22:403–434, 1976.

[18] I. G. Curtiss and P. J. Staff. *J. Chem. Phys.*, 44:990, 1976.

[19] See,[16] page 385, Eqs. (6.1-29) and (6.1-30), and note that the functions $v(x)$ and $a(x)$ appearing in those equations are defined in Eqs. (6.1-13) to be the same as our functions $A(x)$ and $D(x)$ respectively.

The Reversible Isomerization Reaction

The pair of reactions $S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2$ describes the reversible conversion of two isomeric species S_1 and S_2 into each other. The reaction probability rate constants c_j for these monomolecular channels are numerically equal to the rate constants k_j that appear in the corresponding deterministic reaction rate equations for the species concentrations.[17] The propensity functions and state-change vectors for these reaction channels are

$$a_1(\mathbf{x}) = c_1 x_1, \quad a_2(\mathbf{x}) = c_2 x_2, \tag{14}$$

$$\boldsymbol{\nu}_1 = (-1, +1), \quad \boldsymbol{\nu}_2 = (+1, -1). \tag{15}$$

In the absence of any other reaction channels, the total number of isomers x_T will remain constant in time. This circumstance allows us to eliminate one of the species variables, say the S_2 variable, in favor of the other,

$$X_2(t) = x_\mathrm{T} - X_1(t), \tag{16}$$

and thereby obtain a mathematically simpler univariate problem. In this appendix, we shall derive exact expressions for the mean and variance of $X_1(t) \equiv X(t)$ for the initial condition $X(t_0) = x_0$, where x_0 and x_T may be any two integers satisfying $0 \leqslant x_0 \leqslant x_\mathrm{T}$. We should note that solutions to the full chemical master equation are known for the special cases $x_0 = 0$ and $x_0 = x_\mathrm{T}$.[18]

The process $X(t)$ evolves according to the following dynamical rules: If $X(t) = x$, then in the next infinitesimal time dt, X will increase by 1 or decrease by 1 with the respective probabilities $W_+(x)\,dt$ and $W_-(x)\,dt$, where

$$W_+(x) = c_2(x_\mathrm{T} - x), \quad W_-(x) = c_1 x. \tag{17}$$

This kind of dynamical behavior identifies $X(t)$ as a birth-death type Markov process with stepping probability rate functions $W_+(x)$ and $W_-(x)$. Quite generally for such a process, the

time derivatives of the mean and variance are given by[19]

$$\frac{d\langle X(t)\rangle}{dt} = \langle A(X(t)\rangle\,, \tag{18}$$

$$\frac{d\,\mathrm{var}\{X(t)\}}{dt} = 2\left(\langle X(t)\,A(X(t)\rangle - \langle X(t)\rangle\,\langle A(X(t)\rangle\right) + \langle D(X(t)\rangle\,, \tag{19}$$

where

$$A(x) \equiv W_+(x) - W_-(x), \quad D(x) \equiv W_+(x) + W_-(x). \tag{20}$$

For the stepping functions (17), A and D are easily calculated to be

$$A(x) = c_2 x_{\mathrm{T}} - (c_1 + c_2)x,$$

$$D(x) = c_2 x_{\mathrm{T}} + (c_1 - c_2)x.$$

When these forms are substituted into Eqs. (18) and (19), we obtain

$$\frac{d\langle X(t)\rangle}{dt} = c_2 x_{\mathrm{T}} - (c_1 + c_2)\langle X(t)\rangle, \tag{21}$$

$$\frac{d\,\mathrm{var}\{X(t)\}}{dt} = -2(c_1 + c_2)\,\mathrm{var}\{X(t)\} + c_2 x_{\mathrm{T}} + (c_1 - c_2)\langle X(t)\rangle. \tag{22}$$

The time-evolution equation (21) for the mean $\langle X(t)\rangle$ is mathematically identical to the associated deterministic reaction rate equation, although expressed here in terms of the molecular populations instead of concentrations. This is not so in general, but it is the case whenever the propensity functions are linear in the species variables. Equation (21) has the form of the first order linear differential equation $dy(t)/dt = k\,y(t) + f(t)$, for which the general solution in quadrature form is

$$y(t) = \mathrm{e}^{k(t-t_0)}\left\{y(t_0) + \int_{t_0}^{t} f(t')\,\mathrm{e}^{-k(t'-t_0)}dt'\right\}, \tag{23}$$

as may readily be verified by direct differentiation. Evaluating this quadrature form for Eq. (21) using the initial condition $\langle X(t_0)\rangle = x_0$ gives

$$\langle X(t)\rangle = x_0 + (x_\infty - x_0)\left(1 - \mathrm{e}^{-(c_1+c_2)(t-t_0)}\right), \tag{24}$$

where

$$x_\infty \equiv \frac{c_2\, x_{\rm T}}{c_1 + c_2}. \tag{25}$$

By substituting the result (24) into Eq. (22), we obtain for the variance a differential equation that is once again of the first order linear form. When we evaluate the corresponding quadrature solution (23) using the initial condition $\mathrm{var}\{X(t_0)\} = 0$, we get

$$\begin{aligned} \mathrm{var}\{X(t)\} = {} & \frac{c_1\, x_\infty}{(c_1 + c_2)} \left(1 - \mathrm{e}^{-2(c_1+c_2)(t-t_0)}\right) \\ & + \left(\frac{(c_1 - c_2)(x_0 - x_\infty)}{(c_1 + c_2)}\right) \left(\mathrm{e}^{-(c_1+c_2)(t-t_0)} - \mathrm{e}^{-2(c_1+c_2)(t-t_0)}\right). \end{aligned} \tag{26}$$

We note in passing that Eqs. (24) and (26) imply the asymptotic results

$$\langle X(\infty)\rangle = x_\infty \text{ and } \mathrm{var}\{X(\infty)\} = \frac{c_1\, x_\infty}{(c_1 + c_2)}. \tag{27}$$

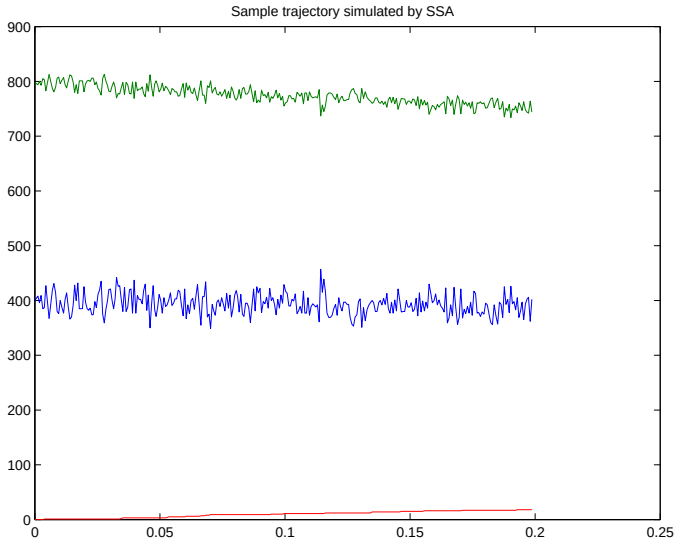

Figure 1: Sample trajectories for Example 1, simulated by the exact SSA. The upper curve is x_2, the middle curve is x_1, and the lower curve is x_3.

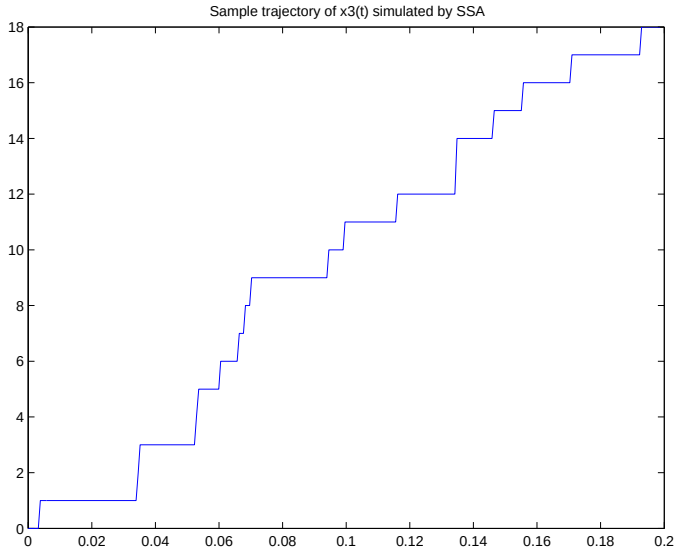

Figure 2: Sample trajectory $x_3(t)$ in Example 1, simulated by SSA.

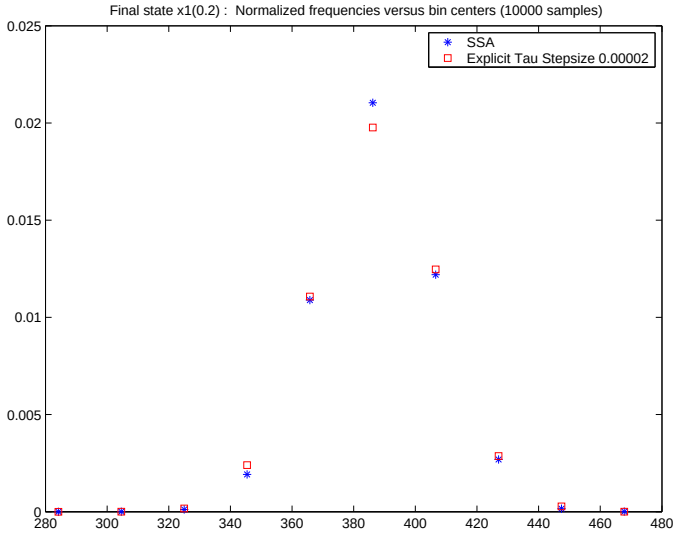

Figure 3: Final state histogram for x_1 in Example 1, computed by the SSA (stars) and the explicit-tau method (squares), the latter with constant stepsize 2×10^{-5}.

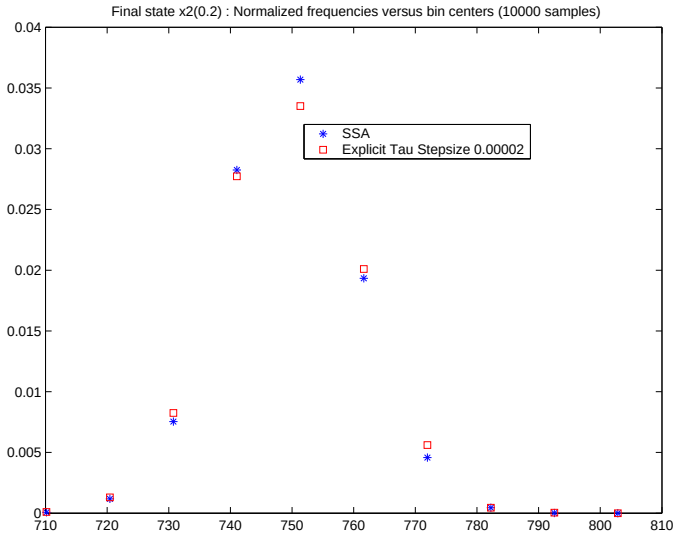

Figure 4: Final state histogram for x_2 in Example 1, computed by the SSA (stars) and the explicit-tau method (squares), the latter with constant stepsize 2×10^{-5}.

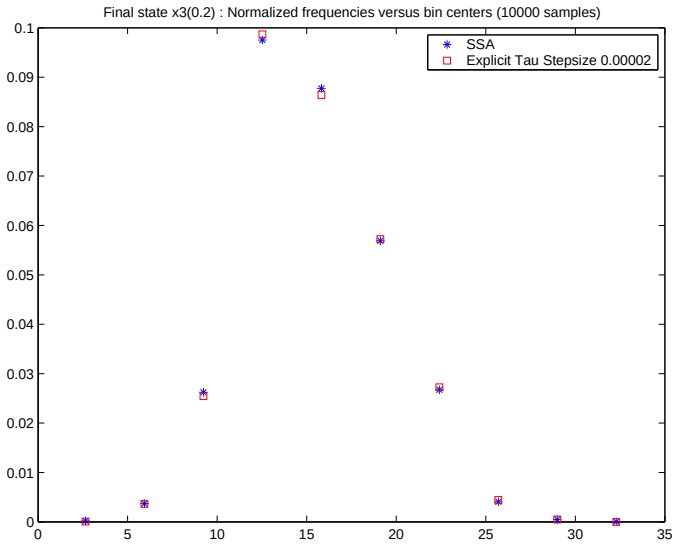

Figure 5: Final state histogram for x_3 in Example 1, computed by the SSA (stars) and the explicit-tau method (squares), the latter with constant stepsize 2×10^{-5}.

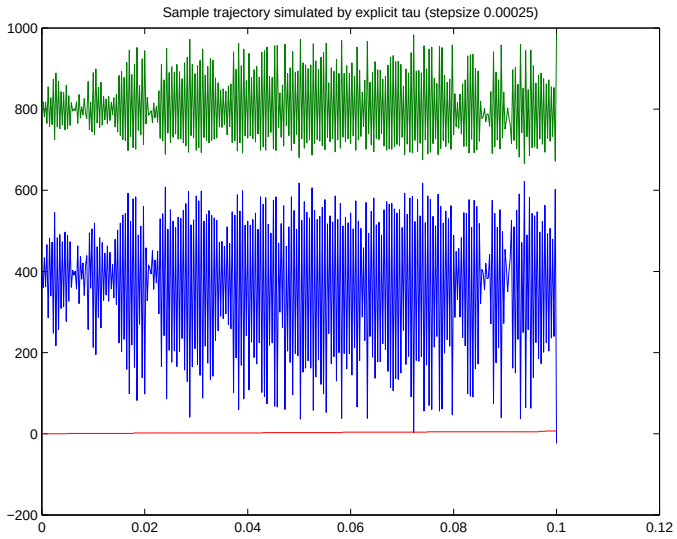

Figure 6: Sample trajectories for Example 1 as simulated by the explicit-tau method with stepsize 2.5×10^{-4}. The trajectories develop unstable oscillations, and yield unrealistic negative states beyond $t \approx 0.1$.

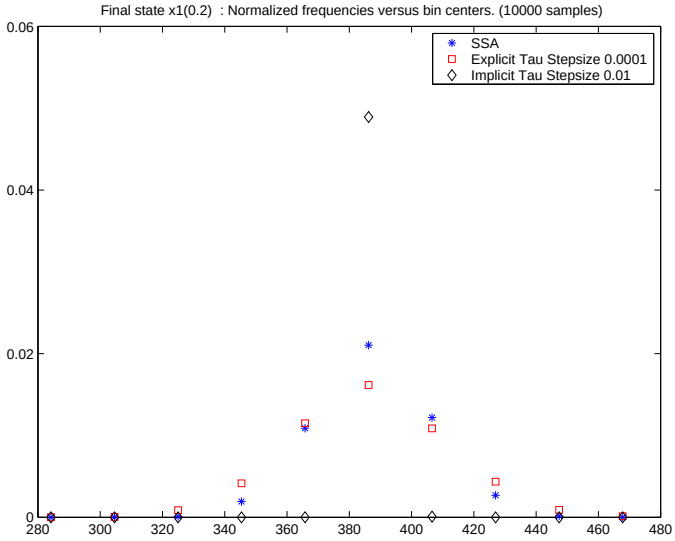

Figure 7: Final state histogram for x_1 in Example 1, as computed by the SSA (stars), the explicit-tau method with stepsize 1×10^{-4} (squares) and the implicit-tau method with stepsize 0.01 (diamonds). Note that the explicit-tau method overestimates the noise while the implicit-tau method underestimates it.

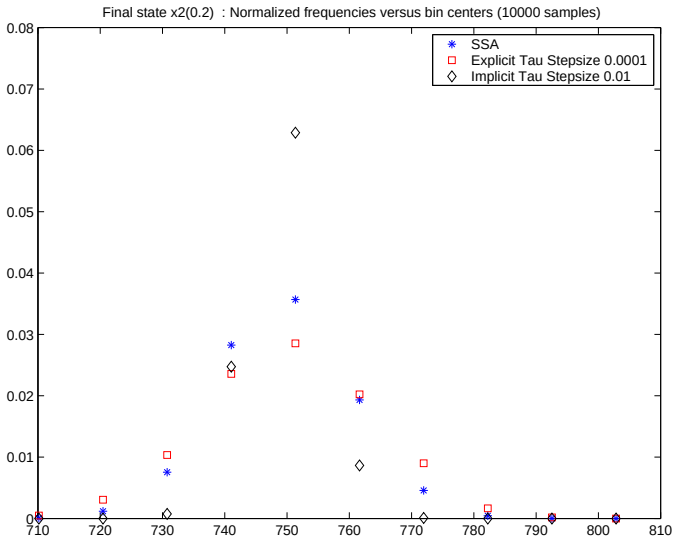

Figure 8: Final state histogram for x_2 in Example 1, as computed by the SSA (stars), the explicit-tau method with stepsize 1×10^{-4} (squares), and the implicit-tau method with stepsize 0.01 (diamonds). Note that the explicit-tau method overestimates the noise while the implicit-tau method underestimates it.

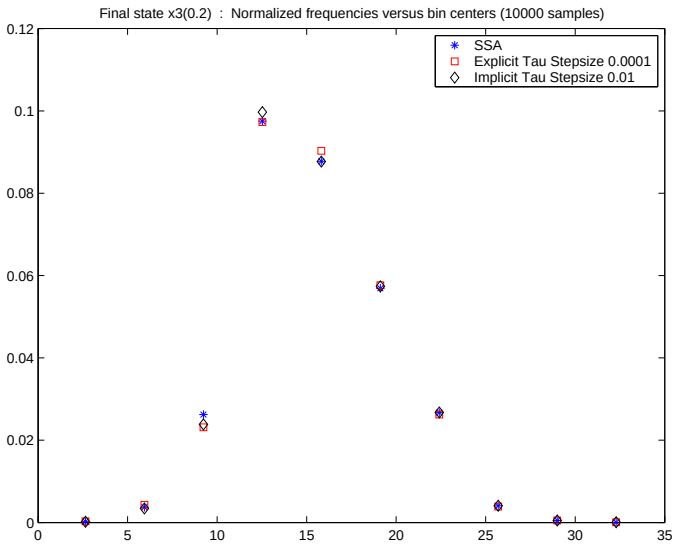

Figure 9: Final state histogram for x_3 in Example 1, as computed by the SSA (stars), the explicit-tau method with stepsize 1×10^{-4} (squares), and the implicit-tau method with stepsize 0.01 (diamonds).

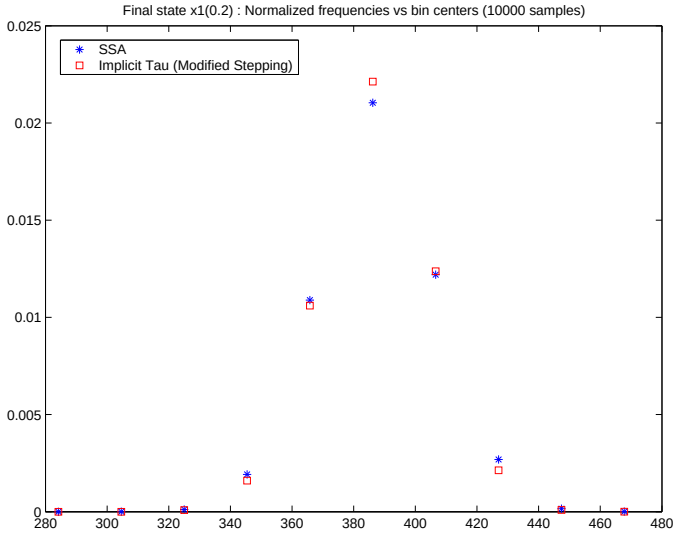

Figure 10: Final state histogram for x_1 in Example 1, as computed by the SSA (stars), and the implicit-tau method (squares) with interlaced stepping.

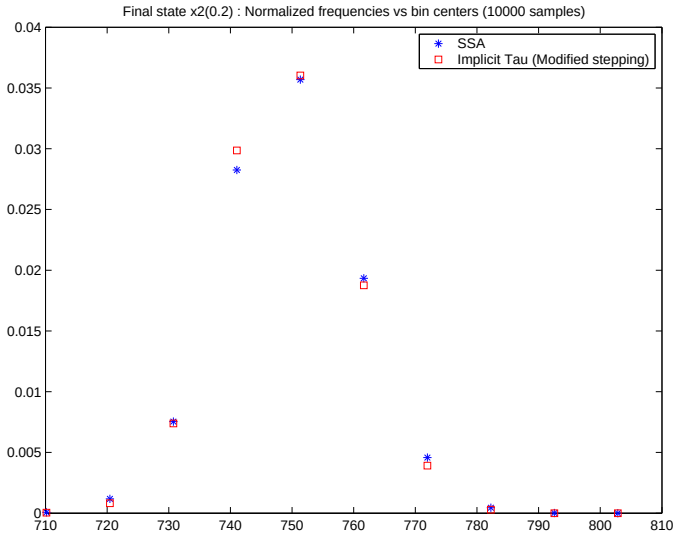

Figure 11: Final state histogram for x_2 in Example 1, as computed by the SSA (stars), and the implicit-tau method (squares) with interlaced stepping.

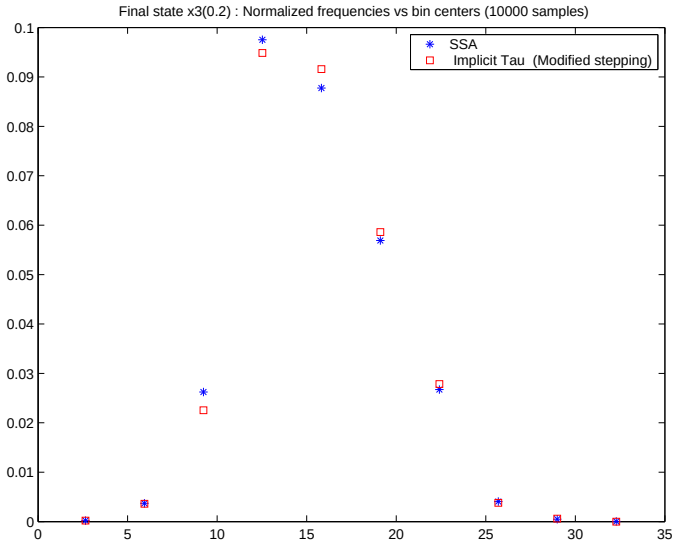

Figure 12: Final state histogram for x_3 in Example 1, as computed by the SSA (stars), and the implicit-tau method (squares) with interlaced stepping.

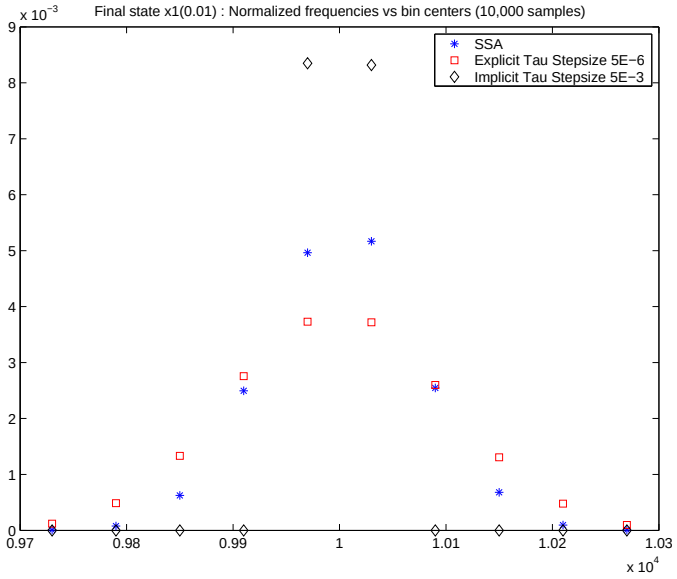

Figure 13: Final state histogram for x_1 in Example 2, as computed by the SSA (stars), the explicit-tau method with step size 5×10^{-6} (squares), and the implicit-tau method with stepsize 5×10^{-3} (diamonds). Note that most of the variation (as computed by SSA) is within 1% deviation from the mean value of 10^4, and thus the noise in this variable may be regarded as negligible. The plot of the histogram has been scaled to fit the narrow range of values of x_1.

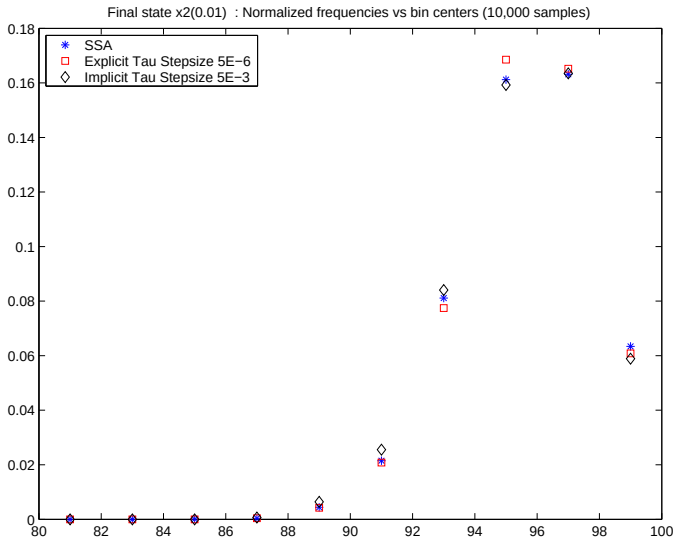

Figure 14: Final state histogram for x_2 in Example 2, as computed by the SSA (stars), the explicit-tau method with step size 5×10^{-6} (squares), and the implicit-tau method with stepsize 5×10^{-3} (diamonds). In contrast to the fluctuations in x_1, the fluctuations in this variable are not negligible.